

CMPE 211 GAME DEVELOPMENT PROJECT

Design Report

Group Members:

Berkay Dursun

Elif Alptekin

Gökhan İrtem

Table Of Content

Class Diagrams………………………………………………………………………………..3

Descriptions of Data…………………………………………………………………………..3

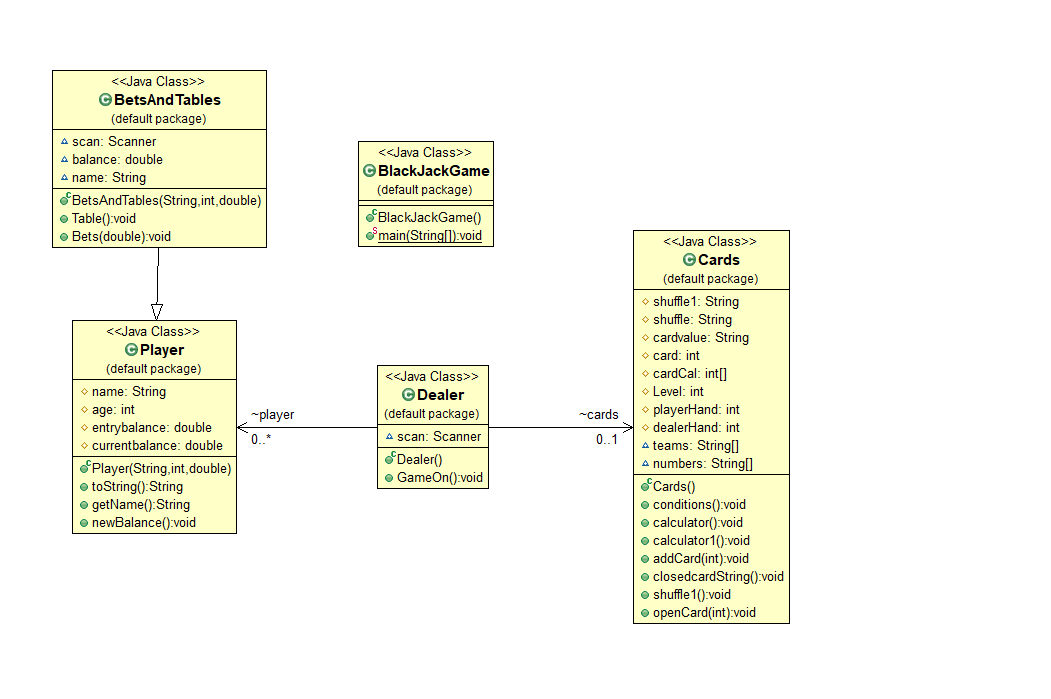
[Descriptions of Methods 5](#_Toc67765284)

[Relationships Between Classes](#_Toc67765285) 7

[Workload Division](#_Toc67765286) 7

Project Name: Blackjack

CLASS DIAGRAMS



DESCRIPTION OF DATA

* In Players class we have a constructor with public which have 3 parameters; name, age, entrybalance.
* Since we extend Player Class to BetsAndTables Class we used this relationship to equalizing data that are name and balance. This reference in the parameter is used to differentiate the instance variables of a class from method parameters with the same names.
* Currentbalance value set as double and used to update players balance.
* We created the balance value as a double to equalize the balance information we got from the player class and use it in our methods.
* Cards Class includes several data in protected modifier to be accessible for subclasses.
* Shuffle1 is used to get the random card as a String value from the numbers array.
* Shuffle is used to get the random card as a String value from the teams array.
* Cardvalue is used equalize to shuffle1.
* Card integer created as auxiliary value in opencard to convert String values into integer values.
* CardCal array is used to store integer values of cards.
* Level is used to equalize index values.
* Playerhand and Dealerhand are used to show calculated values of players and dealers cards.
* In Cards Class we have two String arrays to representing the card of a deck. We divide deck according to categories as teams and numbers.

DESCRIPTIONS OF METHODS

* Table() method from BetsAndTables class: In this method, we planned to adjust the tables according to the players balance.
* Bets() method from BetsAndTables class: In this method, we planned to take the player’s bets in correct conditions.
* conditions() method from Cards class: In this method, we planned to write general conditions about in-game situations.
* calculator() method from Cards class: In this method, we planned to calculate player’s first and second card and dealer’s first card because dealer’s second card must be closed in this step.
* calculator1() method from Cards class: In this method, we planned to calculate dealer’s first card and closed card. Also we showed player’s hand.
* addCard() method from Cards class: In this method, we planned to increase the length of array which we collect the cards as integer and add the last drawn card to the array.
* closedcardString() method from Cards class: In this method, we planned to take dealer’s random closed card and print it.
* shuffle1() method from Cards class: In this method, we planned to take random card and print it.
* openCard() method from Cards class: In this method, we planned to transform the cards from string to integer values.
* GameOn() method from Dealer class: In this method, we planned to run the game as dealer.
* Player() consturctor method from Player class: In this method, we take our players’ name, age and entrybalance informations.
* toString() method from Player class: In this method, we planned to print the informations about players’ name, age and entrybalance.
* newBalance() method from Player class: In this method, we planned to calculate the new balance of our players.
* GameOn() method in BlackJackGame class(Main): In this method we planned to run the game.

RELATIONSHIPS BETWEEN CLASSES

* We extend Player class in BetsAndTables class, to use the informations and values to write the correct code for methods.
* We create an array object called player in the Dealer class to establish a connection between Player class and Dealer class which is used for store our informations and values.
* We create an object called cards in the Dealer class to establish a connection between Cards class and Dealer class which is used for the methods and run the game according to these methods.
* We create an object called dealer in BlackJackGame class(Main) to establish a connection between Dealer class and BlackJackGame class which is used to run the game.

WORKLOAD DIVISION

In our project we aim to make a game while doing that we will make classes and Graphical user interface. We generally aim to seperate the parts as classes to be made by Berkay Dursun and Elif Alptekin and Graphical user interface to be made by Gökhan İrtem but we planned as work on project together all the time.